Miniconf, 2016-12-19

Snippet Learning
(via Word Vectors)

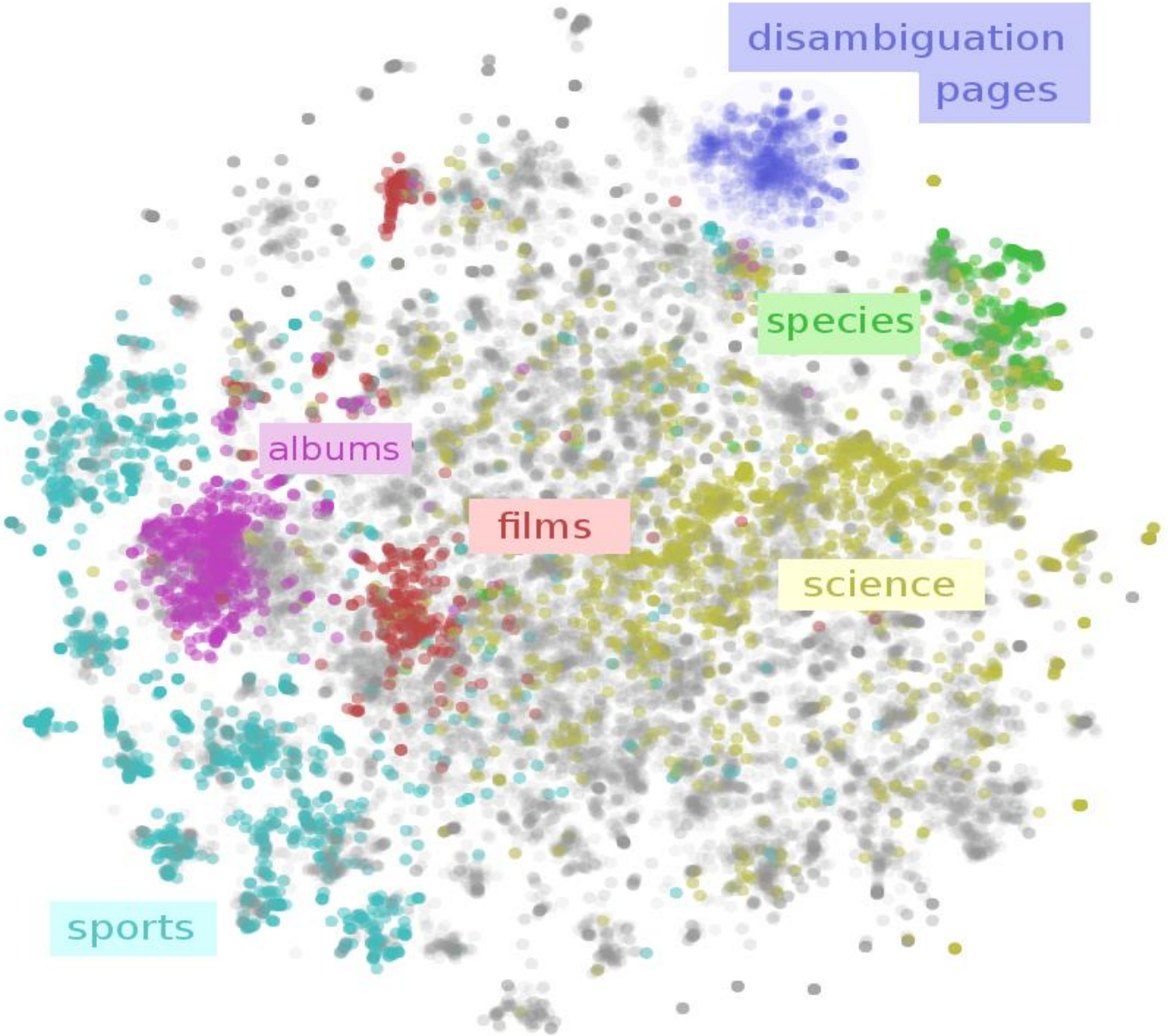
Bernd Malle, PhD Student
Nicola Giuliani, MSc Student

b.malle@hci-kdd.org
n.giuliani@hci-kdd.org

Introduction / TOC

1. Main idea for Snippet Learning
2. What does word embedding mean
3. Word similarity
4. Word (concept) vectors
5. Why use hashtags for such experiments?
6. Hashtag clustering
7. Building snippet descriptors
8. Experimental workflow
9. Future work

Word Vector Clustering / Classification



https://colah.github.io/posts/2015-01-Visualizing-Representations/

Overview / Contents

- General idea: Extracting meaning from small amounts (=snippets) of text => Snippet Learning
- General problem: not enough text for traditional, frequency based methods (TF-IDF, LSA, LDA)
- General approach: Using word / concept vectors trained on large corpora of text and apply them to text snippets, combining them to snipped descriptors
- Experimental approach: Train word vectors on Wikipedia (sub)corpora and apply them to tweets.

What does word embedding mean?

Any technique for mapping a word from its original high-dimensional input space (the set of all words in a language) to a lower-dimensional numerical vector space – in this case to a space of concepts

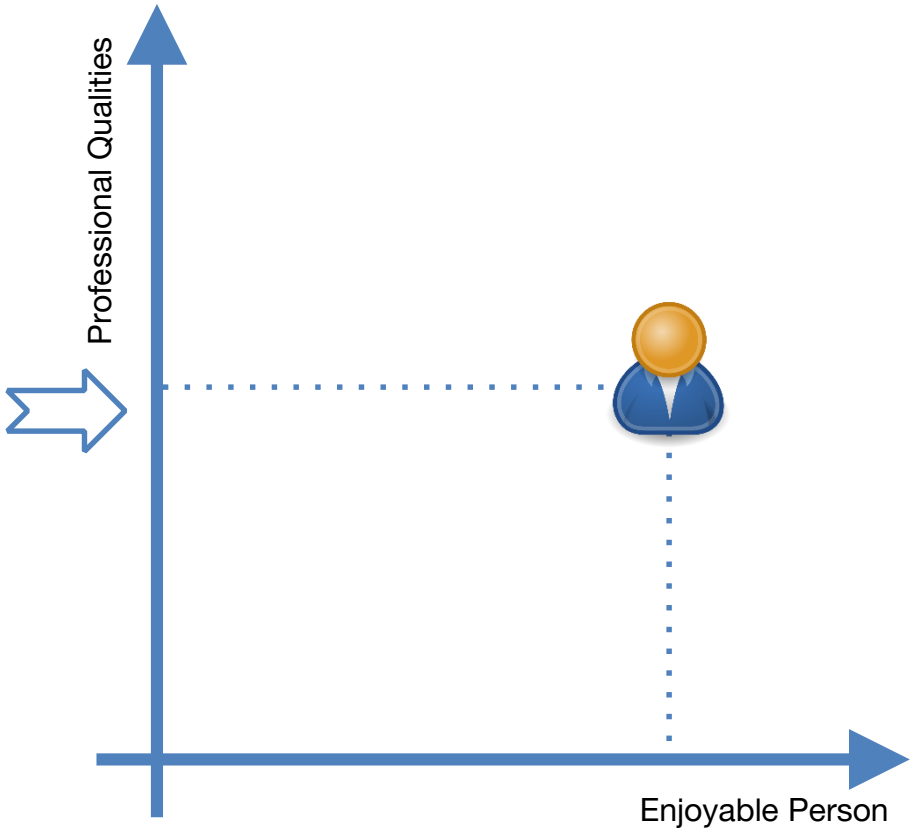


Diagram: https://www.linkedin.com/pulse/you-complete-project-manager-randall-randy-englund

Three main ideas (to take home)

- Similarity in meaning should equal similarity in vectors

  mathematics should be able to encode meaning

- "You shall know a word by the company it keeps" ;)

  The environment of a word gives meaning to it

- Use BIG datasets (millions of billions to words)

  especially neural models require lots of data!

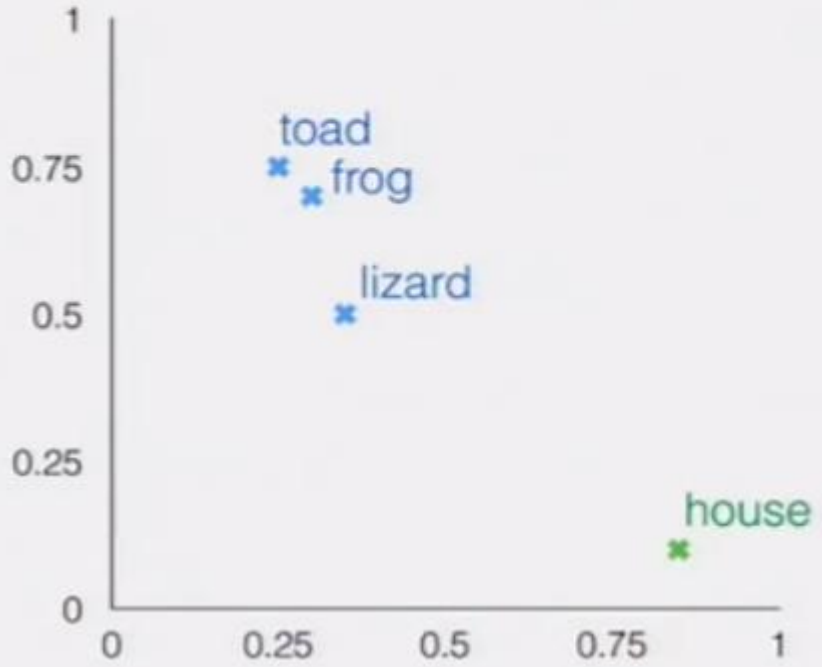
Word Vectors General Assumptions

- The more often 2 words co-occur, the closer their vectors will be
- 2 words have close meanings if their local neighborhoods are similar
- Maps of words (trained on the same dataset) should be similar for each language => can in theory be used for automatic translation…
  - Like a compiler which builds an internal representation of one language (AST) and outputs another language

Distributed representations

Word vectors aren't guaranteed to encode any linguistic relationships between words, but many models produce vectors that do

frog	[0.30 0.70]
toad	[0.25 0.75]
lizard	[0.35 0.50]
house	[0.85 0.10]

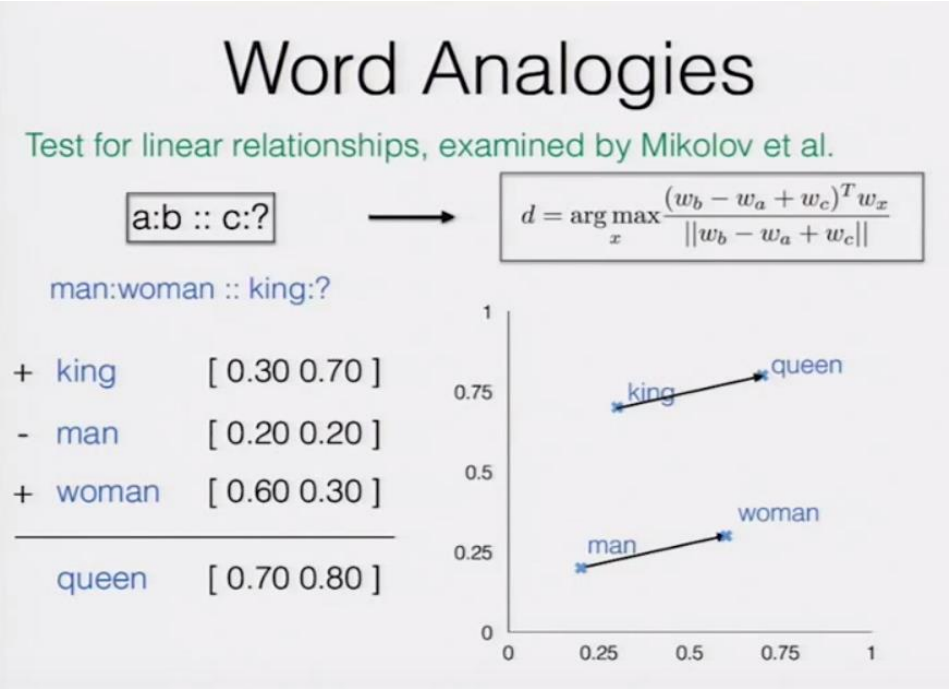


Source: https://www.youtube.com/watch?v=RyTpzZQrHCs

Word vector operations

- Arithmetic

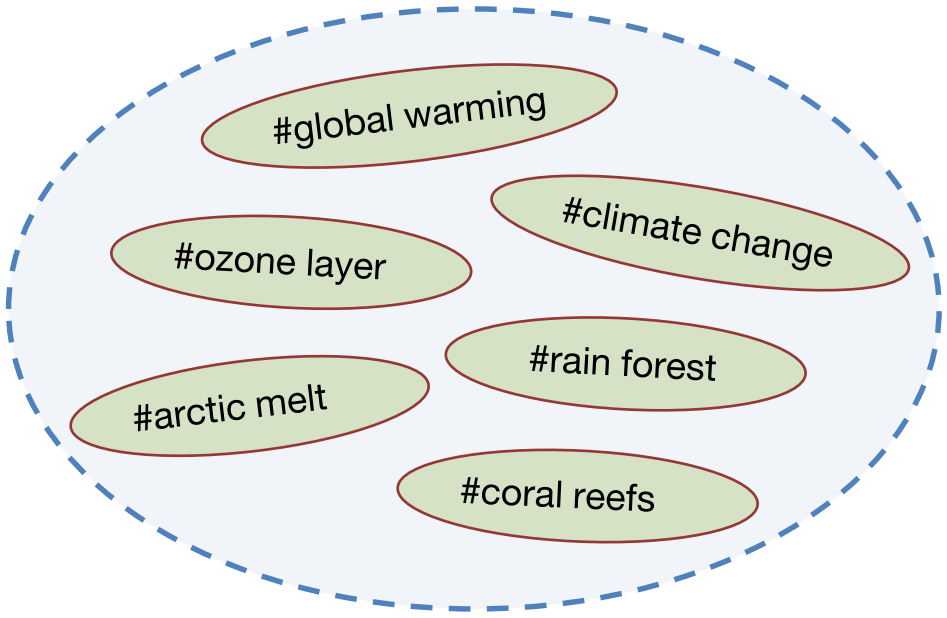
"king - man + woman = queen"



Source: https://www.youtube.com/watch?v=RyTpzZQrHCs

- 2. Nearest Neighbors
  - frog: toad, litoria, lizard, ...
  - works even with numbers (given certain context, like gene sequence)
- 3. Find words that do not belong
  - dog, cat, mouse, fruit basket

Tweets => Hashtags => User labels



Why are tweets ideal to experiment with?

- User provide labels (ground truths) by using hashtags
- Hashtags also imply clustering
- But sometimes we need to pre-process hashtags via word similarity / word sense disambiguation manually
- In our case via WordNet

Hashtag clustering workflow

- Basic idea
  - Use knowledge-based measures that quantify semantic relatedness of words using a semantic network
  - Many measures available: Wu and Palmer metric, Resnik metric (Information Content), …
- Problem
  - Hashtags are “more” than words. They can contain several words, special characters, etc. -> we can not use a measure on this kind of hashtags as their corresponding words would not be contained in wordnet
  - Named Entities (Names, Places, Organizations etc.) are not included in WordNet
- Solution: Preprocessing + Dictionaries

Meng, Lingling, Runqing Huang, and Junzhong Gu. "A review of semantic similarity measures in wordnet." *International Journal of Hybrid Information Technology* 6.1 (2013): 1-12.

Hashtag clustering workflow

Preprocessing - What aspects to consider?

- Case sensitivity -> #Pray := #pray
- Special characters are often used in hashtags BUT
- they often do not add significant meaning
- there is no stable way of handling them
- remove them -> #Pray!! := #Pray
- One #hashtag can contain multiple words
- break the hashtag into its components (words)
  - -> #globalwarming -> global + warming
- solve using dynamic programming

Dictionaries

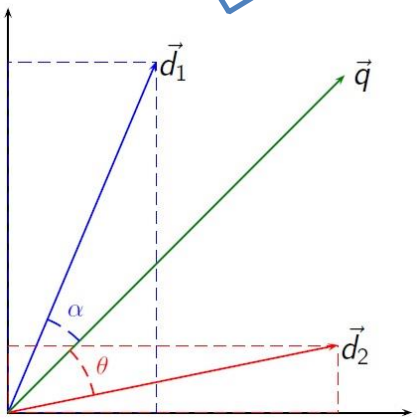
- Use dictionaries for different categories -> names, organizations, etc
- Cluster hashtag according to the dictionary it is found in

Hashtag clustering workflow

Example: #PrayForFu%#$ngHaiti

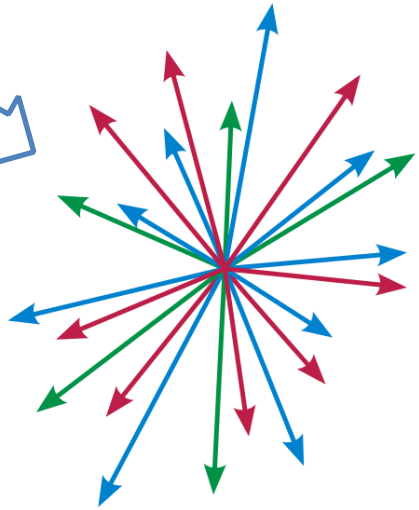
1. to lowercase → prayforfu%#$nghaiti
2. eliminate special characters → prayforfunghaiti
3. look-up in dictionary → NO RESULT
4. look-up in WordNet → NO RESULT
5. break the string → [pray, for, haiti]
6. POS Tagging for each word → [verb, prep., noun]
7. Choose first Noun → haiti
   i. look up in WordNet → NO RESULT
   ii. look-up in dictionary → Cluster: Places

Wikipedia => Word vectors => descriptors



Individual concept vectors

Snippet descriptor
(accumulation, same dimension)



So how to build text snippet representations from vectors?

- First find concept vector for each word (filter out stop-words etc. before)
- Then construct a composite descriptor using some mapping function
- Can be simple (matrix form) or rather complex (convex hull in vector space???) Open question!

Current experimental outline

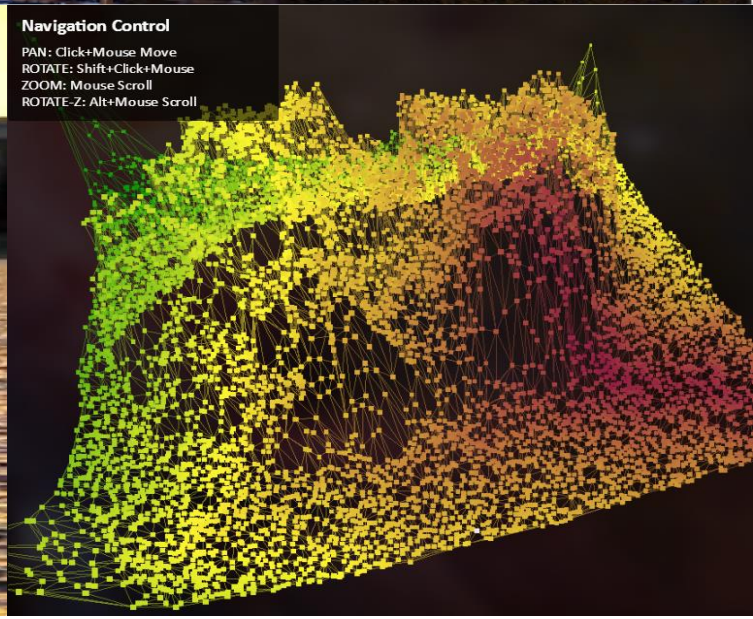
- Cluster the tweets via hashtags as well as word / concept vectors
- Compare the two clusterings obtained and see how much they differ (metric yet to be chosen)
- Apply different descriptors (matrix / convex hull etc.) to tweets and observe the changes in clustering.
- Train the underlying word vector collections on different topical text corpora (Wikipedia history vs. scientific texts).
- Use all of the settings above to run classifiers on individual tweets, comparing their performance to existing methods

Future work…

Use snippet learning methods / concept vectors as building blocks in more complex scenarios:

- Take a Todo List & predict the underlying goal
- Take messages on a board (like snapchat) and predict the underlying project structure
- Sequence prediction: Take any of the two above and output not only the goal, but recommend an optimal strategy to achieve it
- combine with graph recommenders and grow rich……..

Plans for 2017…



Thank you!